Piotr Kużdżał, Inżynieria Obliczeniowa, nr indeksu 286113

„Podstawy Sztucznej Inteligencji”

Scenariusz 4

Temat ćwiczenia: *Uczenie sieci regułą Hebba.*

1. Model neuronu Hebba – zmiana wagi w_{ij} neuronu odbywa się proporcjonalnie do iloczynu jego sygnału wejściowego oraz wyjściowego:

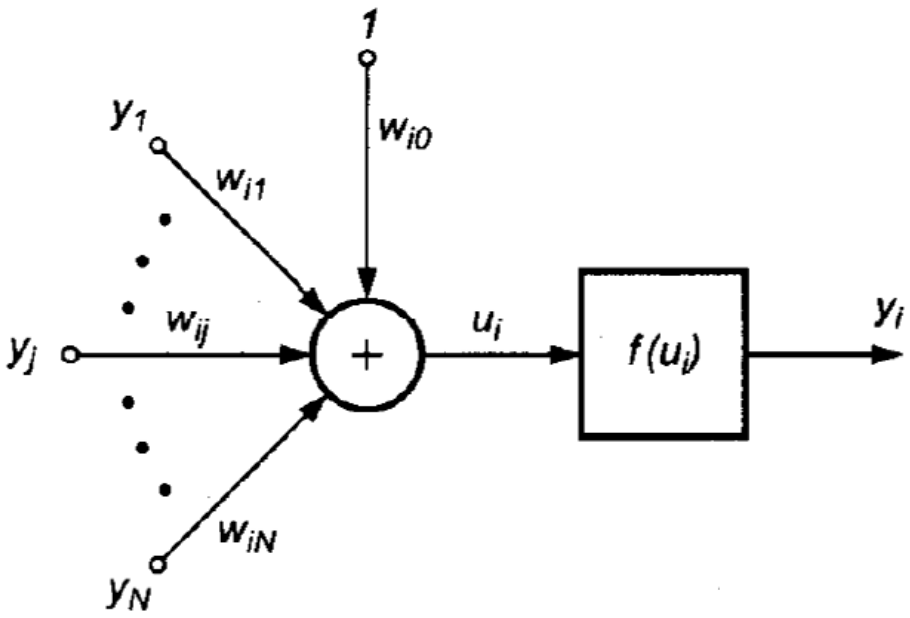
$$\Delta w_{ij} = \eta\, y_j\, y_i$$

gdzie n jest stałą uczenia z przedziału (0,1). Reguła Hebba może być stosowana do różnego typu struktur sieci neuronowych i różnych funkcji aktywacji zastosowanych w modelu neuronu. Reguła Hebba jest praktyczną realizacją stwierdzenia z zakresu neurobiologii:

„Jeżeli akson komórki A bierze systematycznie udział w pobudzeniu komórki B powodującym jej aktywację, to wywołuje to zmianę metaboliczną w jednej lub obu komórkach, prowadzącą do wzrostu skuteczności pobudzania B przez A”.

Uczenie neuronu z zastosowaniem reguły Hebba może się odbywać w trybie bez nauczyciela lub z nauczycielem. W pierwszym przypadku używa się aktualnej wartości y_i sygnału wejściowego neuronu. W uczeniu z nauczycielem wartość sygnału wyjściowego y_i zastępuje się wartością zadana di dla tego neuronu. Regułę Hebba można wówczas zapisać w postaci:

$$\Delta w_{ij} = \eta\, y_j\, d_i$$



Rys. 2.13. Ogólny model neuronu Hebba

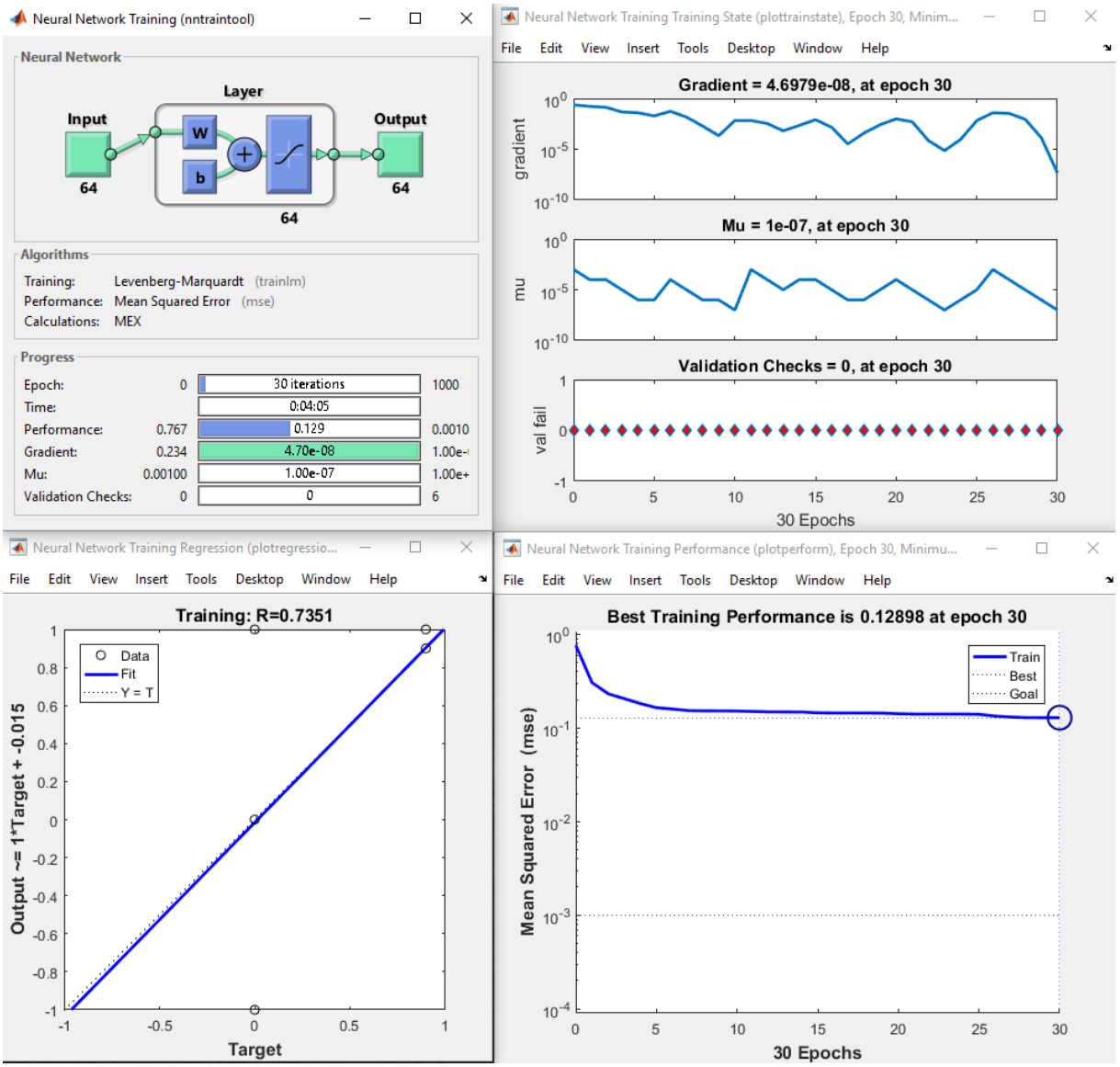
2. Listing programu

```
close all; clear all; clc;

PR=[0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1;
    0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1;
    0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1;
    0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1;
    0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1;
    0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1;
    0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1;
    0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1];
%wejscia do sieci i min/max wartosci wejsc
S=64; %ilosc neuronow na wyjsciu
net = newff(PR,S,{'tansig'},'trainlm','learnh');


    %wejscie
 WE=[0 0 0 0
     0 0 0 0
     1 1 0 1
     1 1 0 1
     1 1 0 1
     1 1 0 1
     0 0 0 0
     0 0 0 0
     0 0 0 0
     1 1 0 1
     0 0 1 0
     0 0 0 0
     0 0 0 0
     0 0 1 0
     1 1 0 1
     0 0 0 0
     1 1 0 1
     0 0 0 0
     1 1 1 1
     0 0 0 0
     0 0 0 0
     1 1 1 1
     0 0 0 0
     1 1 0 1
     1 1 0 1
     0 0 0 0
     0 0 0 0
     0 0 0 0
     0 0 0 0
     0 0 0 0
     0 0 0 0
     1 1 0 1
     1 1 0 1
     0 0 0 0
     1 0 0 0
     0 1 0 1
     0 1 0 1
     1 0 1 0
     0 0 0 0
     1 1 0 1
     1 1 0 1
     0 0 0 0
     0 0 0 1
     1 1 1 0
     1 1 1 0
     0 0 0 1
     0 0 0 0
     1 1 0 1
     0 0 0 0
```

```
        0 0 0 0
        1 1 0 1
        0 0 0 0
        0 0 0 0
        0 0 0 0
        0 0 1 0
        1 1 0 1
        0 0 0 0
        0 0 0 0
        0 0 0 0
        1 1 0 1
        1 1 0 1
        1 1 0 1
        1 1 0 1
        0 0 0 0
        0 0 0 0
        ];
       %Wyjscie
WY=[1 0 0 0 %usmiech
    0 1 0 0 %szok
    0 0 1 0 %pocalunek
    0 0 0 0]; %smutek

lp.dr = 0.5; %wsp zapominania
lp.lr = 0.9; %wsp uczenia
hebRule=learnh([],WE,[],[],WY,[],[],[],[],[],lp,[]);
heb=hebRule';

net.trainParam.epochs = 1000; %ilosc epok
net.trainParam.goal = 0.001; %Cel wydajności
net.trainParam.lr=0.5; % wskaźnik uczenia sie
net=train(net, WE, heb); %trenowanie
%emotikony
usmiech = [0 0 1 1 1 1 0 0;
           0 1 0 0 0 0 1 0;
           1 0 1 0 0 1 0 1;
           1 0 0 0 0 0 0 1;
           1 0 1 0 0 1 0 1;
           1 0 0 1 1 0 0 1;
           0 1 0 0 0 0 1 0;
           0 0 1 1 1 1 0 0];

szok = [0 0 1 1 1 1 0 0;
        0 1 0 0 0 0 1 0;
        1 0 1 0 0 1 0 1;
        1 0 0 0 0 0 0 1;
        1 0 0 1 1 0 0 1;
        1 0 0 1 1 0 0 1;
        0 1 0 0 0 0 1 0;
        0 0 1 1 1 1 0 0];

pocalunek = [0 0 0 0 0 0 0 0;
             0 0 1 0 0 1 0 0;
             0 0 1 0 0 1 0 0;
             0 0 0 0 0 0 0 0;
             0 0 0 0 0 1 0 0;
             0 0 0 1 1 0 0 0;
             0 0 0 0 0 1 0 0;
             0 0 0 0 0 0 0 0];
```

```
124
125 -    smutek = [0 0 1 1 1 1 0 0;
126 -              0 1 0 0 0 0 1 0;
127 -              1 0 1 0 0 1 0 1;
128 -              1 0 0 0 0 0 0 1;
129 -              1 0 0 1 1 0 0 1;
130 -              1 0 1 0 0 1 0 1;
131 -              0 1 0 0 0 0 1 0;
132 -              0 0 1 1 1 1 0 0];
133 -
134 -
135 -    efekt1=sim(net, testa);%symulacja
136 -    efekt=Hebb;
137 -
138 -    disp('Hebb:')
139 -    disp('Usmiech'),disp(sum(efekt(1,':')));
140 -    disp('Szok'),disp(sum(efekt(2,':')));
141 -    disp('Pocalunek'),disp(sum(efekt(3,':')));
142 -    disp('Smutek'),disp(sum(efekt(4,':')));
```

3. Wygenerowane zostały dane uczące i testujące zawierające 4 różne czarno białe emotikony Kolejnym krokiem było uczenie sieci dla różnych współczynników uczenia i zapominania oraz testowanie sieci. Wyniki działania programu znajdują się poniżej.



4. Wnioski
   - Reguła Hebba jest wolniejsza od uczenia z nauczycielem.
   - Duży wpływ ma dobór początkowych wartości wag neuronów.
   - Reguła Hebba pozwala na uczenie sieci bez nauczyciela.